**6.1-Cross Joins and Natural Joins**

1. **Create a cross-join that displays the last name and department name from the employees and departments tables**

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

last\_name VARCHAR2(50),

department\_id NUMBER

);

CREATE TABLE departments (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50)

);

INSERT INTO employees (employee\_id, last\_name, department\_id) VALUES (1, 'Smith', 10);

INSERT INTO employees (employee\_id, last\_name, department\_id) VALUES (2, 'Johnson', 20);

INSERT INTO employees (employee\_id, last\_name, department\_id) VALUES (3, 'Williams', 30);

INSERT INTO departments (department\_id, department\_name) VALUES (10, 'Sales');

INSERT INTO departments (department\_id, department\_name) VALUES (20, 'HR');

INSERT INTO departments (department\_id, department\_name) VALUES (30, 'IT');

SELECT e.last\_name, d.department\_name

FROM employees e

CROSS JOIN departments d;

**Output (Example):**

| **last\_name** | **department\_name** |
| --- | --- |
| Smith | Sales |
| Smith | HR |
| Smith | IT |
| Johnson | Sales |
| Johnson | HR |
| Johnson | IT |
| Williams | Sales |
| Williams | HR |
| Williams | IT |

**2. Create a query that uses a natural join to join the departments table and the locations table. Display the department id, department name, location id, and city.**

CREATE TABLE departments (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50),

location\_id NUMBER

);

CREATE TABLE locations (

location\_id NUMBER PRIMARY KEY,

city VARCHAR2(50)

);

INSERT INTO departments (department\_id, department\_name, location\_id)

VALUES (10, 'Sales', 100);

INSERT INTO departments (department\_id, department\_name, location\_id)

VALUES (20, 'HR', 200);

INSERT INTO departments (department\_id, department\_name, location\_id)

VALUES (30, 'IT', 300);

INSERT INTO locations (location\_id, city)

VALUES (100, 'New York');

INSERT INTO locations (location\_id, city)

VALUES (200, 'Los Angeles');

INSERT INTO locations (location\_id, city)

VALUES (300, 'Chicago');

SELECT department\_id, department\_name, location\_id, city

FROM departments

NATURAL JOIN locations;

**Expected Output:**

| **department\_id** | **department\_name** | **location\_id** | **city** |
| --- | --- | --- | --- |
| 10 | Sales | 100 | New York |
| 20 | HR | 200 | Los Angeles |
| 30 | IT | 300 | Chicago |

**3.Create a query that uses a natural join to join the departments table and the locations table. Restrict the output to only department IDs of 20 and 50. Display the department id, department name, location id, and city**

CREATE TABLE departments (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50),

location\_id NUMBER

);

CREATE TABLE locations (

location\_id NUMBER PRIMARY KEY,

city VARCHAR2(50)

);

INSERT INTO departments (department\_id, department\_name, location\_id)

VALUES (10, 'Sales', 100);

INSERT INTO departments (department\_id, department\_name, location\_id)

VALUES (20, 'HR', 200);

INSERT INTO departments (department\_id, department\_name, location\_id)

VALUES (30, 'IT', 300);

INSERT INTO departments (department\_id, department\_name, location\_id)

VALUES (50, 'Marketing', 400);

INSERT INTO locations (location\_id, city)

VALUES (100, 'New York');

INSERT INTO locations (location\_id, city)

VALUES (200, 'Los Angeles');

INSERT INTO locations (location\_id, city)

VALUES (300, 'Chicago');

INSERT INTO locations (location\_id, city)

VALUES (400, 'San Francisco');

SELECT department\_id, department\_name, location\_id, city

FROM departments

NATURAL JOIN locations

WHERE department\_id IN (20, 50);

**Expected Output:**

| **department\_id** | **department\_name** | **location\_id** | **city** |
| --- | --- | --- | --- |
| 20 | HR | 200 | Los Angeles |
| 50 | Marketing | 400 | San Francisco |